Reference Frame Optimization for Multiple-Path Video Streaming With Complexity Scaling

Gene Cheung, *Senior Member, IEEE*, Wai-tian Tan, *Member, IEEE*, and Connie Chan

***Abstract*—Recent video coding standards such as H.264 offer the flexibility to select reference frames during motion estimation for predicted frames. In this paper, we study the optimization problem of jointly selecting the best set of reference frames and their associated transport QoS levels in a multipath streaming setting. The application of traditional Lagrangian techniques to this optimization problem suffers from either bounded worst case error but high complexity or low complexity but undetermined worst case error. Instead, we present two optimization algorithms that solve the problem globally optimally with high complexity and locally optimally with lower complexity. We then present rounding methods to further reduce computation complexity of the second dynamic programming-based algorithm at the expense of degrading solution quality. Results show that our low-complexity dynamic programming algorithm achieves results comparable to the optimal but high-complexity algorithm, and that gradual tradeoff between complexity and optimization quality can be achieved by our rounding techniques.**

***Index Terms*—Communication systems, optimization methods, video signal processing.**

I. Introduction

ADVANCES in video coding and networking technologies have created many new flexibilities in the design of streaming algorithms. Examples of such flexibilities abound; we focus on two particular ones in this paper. The first flexibility is reference frame selection (RFS) of recent video coding standards such as H.264 [1]. In RFS, each coding block within a predicted frame can choose among a number of previously encoded frames for motion prediction. This allows a live encoder to avoid using lost frames as references, thereby controlling error propagation. The second flexibility is multihoming of clients, where a client may be equipped with multiple network interfaces, such as CDMA2000 and WCDMA. The flexibility of using either or both interfaces has several advantages, including higher throughput with less variability.[1]

While it is clear that streaming can potentially be enhanced by exploiting the aforementioned flexibilities, high complexity is required to jointly select optimal parameters for different options afforded by the many flexibilities. To this end, standard Lagrangian optimization procedures can be employed. Nevertheless, there is no general mechanisms to simultaneously bound the running time of a Lagrangian optimization *and* bound the worst case approximation error. As a result, there are practical challenges in using such optimization schemes for low-latency, quality-guaranteed media delivery. In this paper, we investigate an alternative optimization strategy—applying integer rounding techniques to dynamic programming algorithms. As we will discuss in detail in this paper, this technique generates a solution with bounded complexity and worst case error. Moreover, the strategy is complexity-scalable, where the quality of the obtained approximate solution can be traded off with computation complexity.

The contribution of this paper is twofold. First, we illustrate the aforementioned integer-rounding-based optimization method through an example scenario in which a streaming algorithm jointly optimizes the use of RFS and multiple network interfaces. Specifically, based on feedback information, a live encoder has to choose reference frames based on RFS as well as to transmit a packet one or multiple times, using one or multiple interfaces. The second contribution of this paper is the evaluation of proposed optimization procedure as a practical algorithm. In this regard, comparisons are made with respect to a version of the well-cited optimization framework RaDiO [2].

The remainder of this paper is organized as follows. After discussing related work in Section II, we present in detail our assumptions of source and network models in Section III. We present two optimization algorithms in Section IV: the first algorithm is globally optimal but suffers from high complexity; the second, based on dynamic programming, is locally optimal but has lower complexity. In Section V, we discuss a set of integer-rounding-based procedures to further reduce the complexity of the second developed algorithm at the cost of solution quality. Results and conclusion are presented in Sections VI and VII, respectively.

II. Previous Work

H.264 [1] is a new video coding standard that has demonstrably superior coding performance over previous standards such as MPEG-4 and H.263 over a broad range of bit rates. As part of the new standard is the flexibility of using an arbitrary frame to perform motion estimation, a technique that is originally introduced as Annex N in H.263+ and later as Annex U in H.263++. Early works on optimizing streaming quality using reference frame selection include [3], [4]. Our optimization differs from these recent works by *jointly* selecting reference frame and QoS levels on multiple transmission paths, with the added feature of computation scalability.

Manuscript received December 24, 2004; revised December 19, 2006. This paper was recommended by Associate Editor J. Arnold.

G. Cheung and C. Chan are with Hewlett-Packard Laboratories Japan, Tokyo 168-0072, Japan (e-mail: gene-cs.cheung@hp.com; conniewkchan@yahoo.co.uk).

W. Tan is with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: wai-tian.tan@hp.com).

Digital Object Identifier 10.1109/TCSVT.2007.896620

[1]In some cases, using two transmission paths simultaneously decreases overall performance because of mutual signal interference. We assume here that the paths are orthogonal and therefore additive.

A recent related work [5] reorganized the prediction structure of a group of pictures (GOP) such that the effect of a loss of a single P-frame is minimized. Our work differs from [5] in two regards: 1) while we maximize the expected performance of a GOP—the *average* case, by restructuring the dependencies to minimize the worst effect of all P-frames, [5] minimizes the *worst* case; and 2) [5] performed the restructuring independent of network loss characteristics, while we optimally adopt our scheme to observed network conditions.

A related research topic is multiple description (MD), where video is encoded into two (or more) "descriptions," and each description can be decoded independently of the other(s). For example, an MD stream can be obtained by coding the even frames into stream 0 and coding the odd frames independently from the even frames as stream 1. In [6], it is observed that, when different descriptions are transmitted using different network paths, it is possible to apply error-concealment techniques at the decoder so that drift error due to losses can be greatly reduced. Specifically, such error-concealment techniques can be applied as long as the losses on the different paths are not concurrent. One advantage of the MD scheme is simplicity, since path selection is trivial, and compression can be performed independently of the network conditions. It should be noted that the joint reference frame and QoS level selection on multiple paths subsumes the above MD example as a special case, at the expense of additional computation.

Unlike many previous rate-distortion optimization algorithms [2], [4], [7] which rely on the use of Lagrange multipliers, our optimization is unique in that we use an integer-rounding technique that allows tradeoff between computation complexity with the quality of the obtained solution. This allows us to estimate the quality of the obtained solution given fixed computation resources. Conversely, given a target quality of the solution, we can estimate the amount of resources needed for the tasks.

It should be noted that our dynamic programming plus integer-rounding approach is inspired by classical algorithmic work [8] on the famous NP-hard *knapsack* optimization problem, discussed in detail in [9]. While [8] can be viewed as a starting point, our unique problem requires ingenuity and unique insights in applying and then extending the notion of integer rounding to our more complex objective function.

Among our previous work, we have shown that integer-rounding-based complexity scaling can be applied to reference frame/QoS selection for unipath streaming over QoS-enabled networks [10] and to reference frame/path selection for multipath streaming over best-effort networks [11]. This paper is a noted improvement on our previous work in three important regards: 1) we are simultaneously selecting reference frames and QoS levels on multiple transmission paths; 2) by generalizing to a variable delay model for packet transmission, we incorporate retransmissions into the optimization; and 3) in addition to the previously developed dynamic programming *dimension rounding* technique, to be discussed in Section V-A, a new rounding technique called *index rounding* is introduced in Section V-B, and the two types of rounding techniques are compared and combined in Section V-C.

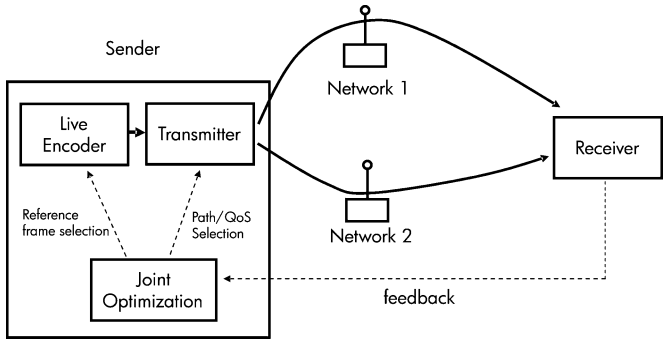


Fig. 1. Application scenario of interest involves live streaming with feedbacks over multiple network interfaces. The sender is responsible for jointly selecting the reference frames and associated QoS levels on multiple transmission paths. An effective scheme to realize such joint optimization is the focus of this paper.

III. Assumptions and Problem Formulation

In this paper, our application scenario of interest is shown in Fig. 1, where a sender is jointly optimizing encoding of video and its transport over two network interfaces. At the live-encoder block, we are choosing previously encoded frames to be used for references. At the transport block, we assume the availability of two network interfaces, each with a number of available QoS levels. There are existing devices with multiple network interfaces already, e.g., 802.11 and GPRS, even though few current applications seek to use them concurrently. There are two motivations to consider two network interfaces. First, it subsumes the more common one-interface case by sending exclusively on one interface. Second, it serves to illustrate how the proposed optimization procedure functions under multiple path scenarios. The remainder of this section is organized as follows. We first present the source and network models, as well as our objective function for optimization. We then discuss some limitations of choices and why they are preferable.

A. Directed Acyclic Graph Source Model

We assume that the optimization in Fig. 1 is run periodically with period P, where, during each optimization instance, an optimization window of M consecutive frames of a $M_{\max}$-frame video sequence is under consideration for (re)transmission, and each frame in the window can be coded either as an intra-coded frame (I-frame) or an inter-coded frame (P-frame). For simplicity of presentation, we will henceforth assume that frames 1 to M are being optimized, though in reality they can be any M consecutive frames in the $M_{\max}$-frame sequence. Each frame $i, F_i, i = 1, \ldots, M$ must be delivered to the client by a playback deadline D_i or be discarded. At the next optimization instance, the optimization window advances k frames in the video sequence where k is the number of leading frames with playback deadlines having expired at the client. Each F_i has a transmission history $\mathbf{h}_i$ which records the times and types of transmission the optimization has selected for F_i in previous optimization instances (more in Section III-B).

Generally, the use of B-frames may improve coding efficiency, but we choose not to include B-frames for two reasons. First, the use of B-frames incurs additional complexity and buffering delay at the client. Second, the baseline profiles of MPEG-4 and H.264 [1] do not include B-frames, meaning

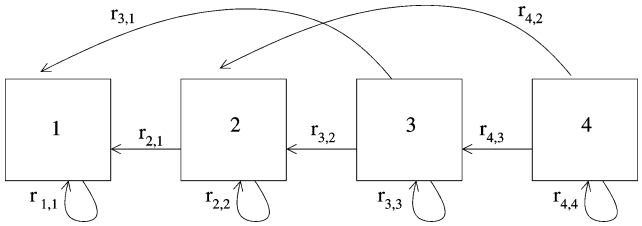


Fig. 2. DAG source model. The quantities $r_{3,3}, r_{3,2}, r_{3,1}$ represent, respectively, the number of bytes for three different choices of coding frame 3, namely, intra-coding, P-frame with frame 2 as reference and P-frame with frame 1 as reference.

that 3GPP-compliant handsets [12], [13] cannot be expected to handle B-frames.

We model the decoding dependencies of the M frames in the optimization window using a directed acyclic graph (DAG) model $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}, |\mathcal{V}| = M$, and edge set $\mathcal{E}$, similar to one used in [2]. Specifically, each frame $F_i, i = 1, \ldots, M$, represented by a node $i \in \mathcal{V}$, has a set of outgoing edges $e_{i,j} \in \mathcal{E}$ to nodes j's. Frame F_i can use frame F_j as reference if and only if $\exists e_{i,j} \in \mathcal{E}$. We define $x_{i,j}$ to be the binary variable indicating whether F_i uses F_j as a reference. Equivalently, given i, we define $x_{i,j}$ as

$$x_{i,j} = \begin{cases} 1, & \text{if } F_i \text{ uses } F_j \text{ as RF} \quad \forall j \in \mathcal{V} | e_{i,j} \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In general, the H.264 syntax allows different coding blocks in a video frame to use different reference frames. In this paper, we restrict all coding blocks in a P-frame to use the same reference frame to reduce complexity of optimization algorithm. The loss in compression efficiency of this assumption is investigated in Section III-D. With this assumption, we have the following *RF constraint*:

$$\sum_{\forall j | e_{i,j} \in \mathcal{E}} x_{i,j} = 1, \qquad \forall i \in \mathcal{V}. \tag{2}$$

We assume that only frames in the past are used for reference, i.e., $\forall e_{i,j} \in \mathcal{E}, i \geq j$. Further, since in practice it is source-coding inefficient to use a reference frame too far in the past, we will limit the number of candidate reference frames for any given predicted frame F_i to be $E_{\max} \ll M$. An example of a DAG model of a four-frame subsequence is shown in Fig. 2 with $E_{\max} = 2$. We denote by $r_{i,j}$ the integer number of bits needed to encode frame F_i if frame F_j is used as reference. This is an approximation since the number of bits depends not only on specific F_j chosen, but also on the reference frame for F_j and so on. A self-referencing arrow for a frame F_i implies an intra-coded frame, and the size of the I-frame is $r_{i,i}$. We assume a sparse *rate matrix* $\mathbf{r}$ of size $O(M^2)$ is computed *a priori* as input to the optimization algorithm (sparse because each row has at most $E_{\max}$ entries). We will discuss how $\mathbf{r}$ is generated for our experiments in Section VI.

B. Network Model

We first assume that the network imposes a maximum transport unit of size MTU bytes, so that a packet of size larger than MTU will be fragmented. For transmission of packet of size fewer than MTU bytes on path k, we assume that a time-invariant packet erasure channel with random delay similar to the one used in [2]. More specifically, let π_k be the packet erasure probability of path k, and let $g_k(\gamma)$ be the shifted Gamma distribution of parameters α_k, λ_k and κ_k that describes the probability distribution of delay random variable γ_k of path k as

$$g_k(\gamma_k) = \frac{\lambda_k^{\alpha_k} (\gamma_k - \kappa_k)^{\alpha_k - 1} e^{-\lambda_k(\gamma_k - \kappa_k)}}{\Gamma(\alpha_k)}, \quad \kappa_k < \gamma_k < \infty \tag{3}$$

where $\Gamma(\alpha_k)$ is the *Gamma function*

$$\Gamma(\alpha_k) = \int_0^\infty \tau^{\alpha_k - 1} e^{-\tau} d\tau, \qquad \alpha_k > 0. \tag{4}$$

This means that a packet sent on path k at time t_o will have probability of correct transmission by time $T, \delta_k(t), t = T - t_o$, which is defined by

$$\delta_k(t) = (1 - \pi_k) \int_{\kappa_k}^{t} g_k(\gamma) d\gamma. \tag{5}$$

On top of the raw transmission path, we assume a set of QoS levels $\mathcal{Q} = \{0, 1, \ldots Q\}$ to improve delivery, either via application-level FEC or simple multiple transmissions. For each frame F_i, we select a QoS level $q0_i$ and $q1_i \in \mathcal{Q}$ for transmission paths 0 and 1, respectively. QoS level $q0_i = q1_i = 0$ denotes the case where F_i is not selected for transmission for the current optimization instance. At a given optimization instance t_o, selection of QoS level $q0_i$ and $q1_i$ and frame size $r_{i,j}$ (resulting from selection of reference frame F_j), together with F_i's transmission history $\mathbf{h}_i$, will induce a *frame delivery success probability* $p(\mathbf{h}_i, q0_i, q1_i, r_{i,j}) \in \mathcal{R}$, where $0 \leq p(\mathbf{h}_i, q0_i, q1_i, r_{i,j}) \leq 1$. There is dependence of $p()$ on $r_{i,j}$ because a large frame size will likely negatively impact the delivery success probability of the entire frame as more data are pushed through the network.

Though the optimality of the algorithms to be developed does not depend on the particular definition of $p(\mathbf{h}_i, q0_i, q1_i, r_{i,j})$, as a concrete case study, we now derive $p(\mathbf{h}_i, q0_i, q1_i, r_{i,j})$ given our network model assuming that $\mathcal{Q}$ consists only of simple multiple transmissions. We first define F_i's history $\mathbf{h}_i$ of length l_i as

$$\mathbf{h}_i = \left\{ \left(f_i, q0_i^{(1)}, q1_i^{(1)}, t_i^{(1)} \right), \left(q0_i^{(2)}, q1_i^{(2)}, t_i^{(2)} \right), \ldots, \left(q0_i^{(l_i)}, q1_i^{(l_i)}, t_i^{(l_i)} \right) \right\} \tag{6}$$

where the reference frame F_j selected for F_i is denoted by f_i,[2] and QoS selections and transmission time of instance k are denoted by $q0_i^{(k)}, q1_i^{(k)}$ and $t_i^{(k)}$, respectively. Let $n_i = \lceil r_{i,f_i} / (8 * \text{MTU}) \rceil$ be the number of packets required to encode F_i using reference frame f_i. The frame delivery failure probability of using QoS $q0_i$ at time t_o is then $\zeta_0(q0_i, n_i, t), t = D_i - t_0$

$$\zeta_0(q0_i, n_i, t) = (1 - \delta_0(t)^{n_i})^{q0_i}. \tag{7}$$

[2]We assume reference frame for a frame F_i is selected only once. Subsequent transmissions of F_i use the same earlier selected reference frame.

$p(\mathbf{h}_i, q0_i, q1_i, r_{i,j})$ can now be written as (8), shown at the bottom of the page.

1) Network Resource Constraint: Like any resource-allocation problems, we impose constraints on the amount of resource we can use, which in this case is the aggregate ability to properly deliver the M frames in the optimization window using QoS until the next optimization instance. Assuming that a QoS assignment $q0_i$ results in a cost of $c(q0_i) \in \mathcal{R}$ per bit, the constraints for path 0 and path 1 are respectively

$$\sum_{i=1}^{M} \sum_{\forall j | e_{i,j} \in \mathcal{E}} x_{i,j} c(q0_i) r_{i,j} \le \bar{R}_0$$
$$\sum_{i=1}^{M} \sum_{\forall j | e_{i,j} \in \mathcal{E}} x_{i,j} c(q1_i) r_{i,j} \le \bar{R}_1. \quad (9)$$

Equation (9) represents a bit-rate constraint per path, where $c(q0_i)$ is the overhead in channel coding or multiple transmissions given QoS level $q0_i$. Constraint parameters $\bar{R}_0$ and $\bar{R}_1$ are network-available bandwidths scaled by optimization period P, where each network bandwidth can be estimated using congestion control algorithms like TCP-friendly rate control (TFRC) [14], so that the total output bits for M-frame time for paths 0 and 1 do not exceed $\bar{R}_0$ and $\bar{R}_1$ bits per optimization instance, respectively. While important, we consider congestion control orthogonal to our reference frame-selection problem, and we will merely assume that an available scheme like TFRC periodically estimates the available network bandwidths on which we perform our optimization.

C. Objective Function

Ideally, the objective function would represent perceptive distortion, the study of which is beyond the scope of this paper. Another commonly employed objective function is the average peak signal-to-noise ratio (PSNR). While computing the actual PSNR between two video sequences is straightforward, the computation incurred by having to compute the actual PSNR for many possible loss patterns and choices for reference frames is prohibitively high. Accurate modeling of PSNR for arbitrary loss patterns is still an area of active research [15], [16]. The objective function we selected instead is the expected number of correctly decoded frames at the decoder. Each frame F_i is correctly decoded if and only if F_i and all frames F_j's it depends on are delivered on-time and drop-free. We write $j \preceq i$ if frame F_i depends on frame F_j. Advantages of this function include being mathematically tractable, and having simple and intuitive interpretation. Mathematically, maximizing this objective function means computing

$$\max_{\{x_{i,j}\},\{q0_i\},\{q1_i\}} \left\{ \sum_{i=1}^{M} \prod_{\forall j \preceq i} \sum_{\forall k | e_{j,k} \in \mathcal{E}} x_{j,k} p(\mathbf{h}_j, q0_j, q1_j, r_{j,k}) \right\}. \quad (10)$$

The problem is then: given precomputed rate matrix $\mathbf{r}$, delivery success probability function $p(\mathbf{h}_i, q0_i, q1_i, r_{i,j})$, and cost function $c(q_i)$, find variables $\{x_{i,j}\}$, $\{q0_i\}$ and $\{q1_i\}$ that maximize (10) while satisfying the integer constraint (1), the RF constraint (2), and the network resource constraints (9). This formally defined optimization is called the RF/QoS/Path selection problem (RQP selection).

TABLE I
CODING COMPARISON OF CHOOSING REFERENCE FRAME ON A PER-BLOCK BASIS (FLEX) AND PER-FRAME BASIS (FIX)

sequence	reference frame	Q_I	Q_P	bitrate	PSNR
mother	flex	22	18	131.28	44.53
mother	fix 1	22	18	137.59	44.45
mother	fix 2	22	18	137.61	44.51
mother	fix 3	22	18	137.44	44.53
mother	fix 4	22	18	136.74	44.52
mother	fix 5	22	18	135.84	44.54
news	flex	25	20	138.60	42.47
news	fix 1	25	20	140.38	42.44
news	fix 2	25	20	141.57	42.46
news	fix 3	25	20	141.72	42.48
news	fix 4	25	20	141.57	42.46
news	fix 5	25	20	141.42	42.44

D. Consequences of Assumed Models

To show that the RF constraint (2) of using a single reference frame for all macroblocks in a P-frame during motion prediction is not excessive in terms of coding efficiency, we compared the rate-distortion performance of a scheme using flexible reference frame motion prediction (`flex`) with a scheme using a single reference frame fixed at frame F_{i-n} for each P-frame F_i (`fix n`). Table I shows the coding performance for the two schemes for MPEG sequences `mother` and `news` using the quantization parameters Q_I and Q_P for I-frames and P-frame at $QCIF$ size. We see that, in general, the PSNRs for the two schemes are almost identical; this is expected since the same quantization parameters were used. What we also observe is that the encoded bit rate for `fix n` deviates from `flex` by at most 4.8% for `mother` and 1.3% for `news`. This shows experimentally that the overhead in rate-distortion performance by imposing the RF constraint (2) on the optimization is justifiably small for the two sequences we use in this paper, even though larger differences are possible for other choices of sequence or QP.

$$= \begin{cases} 1, & \text{if } F_i \text{ is ACKed} \\ 1 - [\zeta_0(q0_i, n_i, D_i - t_o)\zeta_1(q1_i, n_i, D_i - t_o) & \text{o.w.} \\ \prod_{k=1}^{l_i} \zeta_0\left(q0_i^{(k)}, n_i, D_i - t_i^{(k)}\right) \zeta_1\left(q1_i^{(k)}, n_i, D_i - t_i^{(k)}\right)\Big] & \end{cases} \quad (8)$$

function $Sum(i, R_0, R_1, \mathbf{w})$
1. if ($R_0 < 0$) or ($R_1 < 0$) // base case 1
2. { return $-\infty$; }
3. if ($i = 1$) // base case 2
4. { $s := \max_{q0,q1 \in \mathcal{Q} \mid c(q0)r_{1,1} \le R_0,\ c(q1)r_{1,1} \le R_1} p(\mathbf{h}_1, q0, q1, r_{1,1})$;
5. return $s * (1 + w_1) + \sum_{j=2}^{M} w_j$;
6. }
7. $S := 0$; // recursive case
8. for each j such that $e_{i,j} \in \mathcal{E}$,
9. { for each $q0, q1 \in \mathcal{Q}$,
10. { $\mathbf{w}' := \mathbf{w}$;
11. if ($j \neq i$)
12. { $w'_j := w_j + p(\mathbf{h}_i, q0, q1, r_{i,j})(1 + w_i)$;
13. $w'_i := 0$; }
14. else
15. { $w'_j := p(\mathbf{h}_j, q0, q1, r_{j,j})(1 + w_j)$; }
16. $s := Sum(i - 1, R_0 - c(q0)r_{i,j}, R_1 - c(q1)r_{i,j}, \mathbf{w}')$;
17. $S := \max(S, s)$;
18. } }
19. return S;

Fig. 3. Globally optimal $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ in strongly exponential time.

IV. Globally and Locally Optimal Algorithms

It is perhaps not surprising that the RQP selection problem is NP-hard. A proof of NP-hardness, similar to that in [10], is shown in Appendix A. To tackle the problem, our three-step approach is as follows. First, we construct an algorithm $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ that solves the optimization optimally but in *strongly exponential* time. Second, we simplify the algorithm to produce a dynamic-programming-based $\mathrm{Sum}(i, R_0, R_1)$ that is *locally* optimal and *weakly exponential*. Empirical results will be shown later in Section VI that suggest the two algorithms achieve comparable performance despite differences in complexity. Finally, we introduce rounding techniques that allow multiple complexity–quality tradeoff points for $\mathrm{Sum}(i, R_0, R_1)$. Experimental results characterizing the tradeoffs for different rounding techniques will be presented in Section VI.

A. Globally Optimal Algorithm in Strongly Exponential Time

We begin with the development of the globally optimal but strongly exponential time algorithm $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$, shown in Fig. 3. By "strongly," we mean that the running time is independent of the parametric values of the algorithm input. $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ returns the maximum expected number of correctly decoded frames for the M-frame subsequence given that resources R_0 and R_1 are available for F_1 to F_i. $\mathbf{w} \in \mathcal{R}^M$ is the *weight vector* where w_i reveals the potential benefit of correctly decoding F_i—benefit from *dependees* in subset $\{F_{i+1}, \ldots, F_M\}$—in addition to F_i itself. A call to $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1, \mathbf{0})$, where $\mathbf{0}$ is the zero vector of dimension M, would yield the optimal objective value to the RQP selection problem.

For the recursive case (lines 7–18), the algorithm attempts every possible combination of RF (j) and QoS ($q0$ and $q1$), resulting in successful transmission probability $p(\mathbf{h}_i, q0, q1, r_{i,j})$ and budget consumption $c(q0)r_{i,j}$ and $c(q1)r_{i,j}$ for paths 0 and 1, respectively. The crux of the algorithm lies in the weight passing from F_i to RF F_j using the following equation:

$$w'_j := w_j + p(\mathbf{h}_i, q0, q1, r_{i,j})(1 + w_i) \tag{11}$$

Equation (11) essentially states that successful decoding of F_j will reap additional benefits of expected decoding of $F_i, p(\mathbf{h}_i, q0, q1, r_{i,j})$, and F_i's dependees, $p(\mathbf{h}_i, q1, q1, r_{i,j})w_i$. The first base case (lines 1–2) is when one or both of the budget constraints is violated, and the algorithm returns $-\infty$ to signal the violation. The second base case (lines 3–6) is when the root node is reached. Because it has no earlier frame to recurse, the algorithm simply seeks the maximum transmission probability for F_1 in two paths using two leftover budgets R_0 and R_1. At this point, the benefit of each P-frame has been folded into an earlier I-frame that is the root of the prediction, so the algorithm simply returns the sum of benefits from all I-frames (line 5). A proof of optimality is provided in Appendix B.

The complexity of $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1, \mathbf{0})$ can be deduced as follows. The two nested loops in the recursive case have $Q * Q * E_{\max}$ iterations, and each spurts a recursive call. The total number of recursive calls are: $Q^2E_{\max} + (Q^2E_{\max})^2 + \cdots + (Q^2E_{\max})^{(M-1)} \le O((Q^2E_{\max})^M)$. Since each recursive call has at most $Q^2 * E_{\max}$ comparisons in the recursive loop, we can conclude that the complexity of $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1, \mathbf{0})$ is $O((Q^2E_{\max})^{M+1})$.

B. Locally Optimal Algorithm in Weakly Exponential Time

Given that $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ is strongly exponential, it is difficult to reduce its complexity in any formal way. Our approach then is to first simplify it so that it becomes *weakly* exponential—and, hence, implementable in dynamic programming—at the cost of losing global optimality. By "weakly," we mean the running time is exponential only in the size of the algorithm input bits, used to encode parametric values of the input. This is also called *pseudo-polynomial* in some literature [17].

In $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$, local information are passed globally via the weight vector $\mathbf{w}$. If we eliminate weight passing entirely, the algorithm is restricted to local searches and, hence, is locally optimal; this is the idea behind the simplified version. It is composed of two recursive functions, $\mathrm{Sum}(i, R_0, R_1)$ and $\mathrm{Prod}(j, i, R_0, R_1)$. $\mathrm{Sum}(i, R_0, R_1)$ returns the locally optimal expected number of correctly decodable frames for frame F_1 to F_i given R_0 and R_1 network resource units are available for paths 0 and 1, respectively. $\mathrm{Prod}(j, i, R_0, R_1)$ returns the probability that F_j is *decoded* correctly given R_0 and R_1 network resource units of paths 0 and 1 are locally optimally distributed from F_1 to F_i. A call to $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1)$ will yield the locally optimal solution. $\mathrm{Sum}(i, R_0, R_1)$ and $\mathrm{Prod}(j, i, R_0, R_1)$ are shown in Figs. 4 and 5, respectively.

The recursive case (lines 5–15) of $\mathrm{Sum}(i, R_0, R_1)$ is similar to the one in the original $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$; essentially, it locally tests every combination of RF j and QoS $q0$ and $q1$ for F_i for the maximal expected number of decodable frames. For a given selection of RF j and QoS $q0$ and $q1$, it induces a resource expense of $c(q0)r_{i,j}$ and $c(q1)r_{i,j}$ for paths 0 and 1 respectively, and hence a decoding probability for F_i of $p(\mathbf{h}_i, q0, q1, r_{i,j}) * \mathrm{Prod}(j, i-1, R_0 - c(q0)r_{i,j}, R_1 - c(q1)r_{i,j})$. That is added to the expected sum for F_1 to F_{i-1}—the recursive term $\mathrm{Sum}(i - 1, R_0 - c(q0)r_{i,j}, R_1 - c(q1)r_{i,j})$. The base case (lines 3–4) is the same as the first base case in the original $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$.

```
function Sum(i, R_0, R_1)
1.  if  ( DPsum[i, R_0, R_1] is filled )        // DP case
2.  {    return DPsum[i, R_0, R_1];    }
3.  if  ( R_0 < 0 ) or ( R_1 < 0 )              // base case
4.  {    return -∞;    }
5.  S := 0;                                     // recursive case
6.  for each j such that e_{i,j} ∈ E,
7.  {   for each q0, q1 ∈ Q,
8.    { s := Sum(i - 1, R_0 - c(q0)r_{i,j}, R_1 - c(q1)r_{i,j});
9.      if  ( j = i )                           // I-frame
10.     {   s := s + p(h_i, q0, q1, r_{i,j});   }
11.     else                                    // P-frame
12.     {   s := s + p(h_i, q0, q1, r_{i,j}) * Prod(j, i - 1, R_0 -
        c(q0)r_{i,j}, R_1 - c(q1)r_{i,j});    }
13.     if  ( s > S )
14.     {   (S, X, Y, Z) := (s, j, q0, q1);    }
15. } }
16. ( DPsum[i, R_0, R_1], DPind[i, R_0, R_1] ) := ( S, X );
17. ( DPqos0[i, R_0, R_1], DPqos1[i, R_0, R_1] ) := ( Y, Z );
18. return S;
```

Fig. 4. Locally optimal $\mathrm{Sum}(i, R_0, R_1)$ with weakly exponential time.

```
function Prod(j, i, R_0, R_1)
1.  if  ( R_0 < 0 ) or ( R_1 < 0 )          // base case 1
2.  {  return 0;    }
3.  if  ( j = i = 1 )                       // base case 2
4.  {  return DPsum[1, R_0, R_1];    }
5.  X := DPind[i, R_0, R_1];
6.  ( Y, Z ) := ( DPqos0[i, R_0, R_1], DPqos1[i, R_0, R_1] );
7.  if  ( j < i )                           // recursive case
8.    { P := Prod(j, i - 1, R_0 - c(Y)r_{i,X}, R_1 - c(Z)r_{i,X});  }
9.  else                                    // j = i
10. { P := p(h_i, Y, Z, r_{i,X}) * Prod(X, i-1, R_0-c(Y)r_{i,X}, R_1-c(Z)r_{i,X}); }
11. return P;
```

Fig. 5. Companion $\mathrm{Prod}(j, i, R_0, R_1)$ for locally optimal algorithm.

Differing from $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$, the results of this search are stored in the $[i, R_0, R_1]$ entries of the four DP tables, DPsum[ ], DPind[ ], DPqos0[ ], and DPqos1[ ] (lines 16–17). DP tables are lookup tables so that, if the same subproblem is called again, the already computed results can be simply looked up and returned (lines 1-2).

Assuming $\mathrm{Prod}(j, i, R_0, R_1)$ does not introduce further complexity (to be discussed), the complexity of $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1)$ is bounded by the time required to construct the DP tables of dimension $M * \bar{R}_0 * \bar{R}_1$. To fill each entry, we call function $\mathrm{Sum}(i, R_0, R_1)$ as shown in Fig. 4, which has $O(E_{\max} Q^2)$ operations to account for the two `for` loops from lines 6–15 in the recursive case. Therefore, we can conclude that the complexity of $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1)$ is $O(M E_{\max} Q^2 \bar{R}_0 \bar{R}_1)$. Note that the complexity is weakly exponential because $\bar{R}_0$ and $\bar{R}_1$ are encoded in $\lceil \log_2 \bar{R}_0 \rceil$ and $\lceil \log_2 \bar{R}_1 \rceil$ bits as input, respectively. Hence, complexity $O(\bar{R}_0 \bar{R}_1)$ means that the algorithm is exponential in the size of the input.

C. Companion Recursive Function for Locally Optimal Algorithm

From lines 8 and 12 of Fig. 4, we assume that $\mathrm{Prod}(j, i, R_0, R_1)$ is called after $\mathrm{Sum}(i, R_0, R_1)$ has been called, so we will assume entries $[i, R_0, R_1]$ of the DP tables are available during execution of $\mathrm{Prod}(j, i, R_0, R_1)$.

The recursive case has two subcases: 1) when $j < i$ (line 8 of Fig. 5), in which case we recurse on $\mathrm{Prod}(j, i-1, \cdot)$ given that we know resources $c(Y) r_{i,X}$ and $c(Z) r_{i,X}$ on paths 0 and 1 are optimally used for node i; and 2) when $j = i$ (line 10), in which case we know term i of the product term—$p(\mathbf{h}_i, Y, Z, r_{i,X})$. The maximum product will be this term times the recursive term $\mathrm{Prod}(Y, i-1, R_0 - c(Y) r_{i,X}, R_1 - C(Z) r_{i,X})$. The two base cases (lines 1-4) are similar to the two base cases for $\mathrm{Sum}(i, R_0, R_1)$.

Though not written in Fig. 5 for simplicity of presentation, a DP table $\mathrm{DPprod}[j, i, R_0, R_1]$ can be similarly used to store solutions to subproblems to avoid solving the same subproblem twice. Because the number of reference frames is bounded by $E_{\max}$, at most $E_{\max} * M * \bar{R}_0 * \bar{R}_1$ entries of the DP table will be filled. The complexity of $\mathrm{Prod}(j, i, R_0, R_1)$ is also bounded by the time required to fill the $O(E_{\max} * M * \bar{R}_0 * \bar{R}_1)$ necessary entries of the DP table. Since there are no loops in Fig. 5, it takes constant time to fill each entry in the DP table. Hence, the complexity of $\mathrm{Prod}(j, M, \bar{R}_0, \bar{R}_1), \forall j \mathrm{s.t.} E_{i,j} \in \mathcal{E}$, is $O(E_{\max} M \bar{R}_0 \bar{R}_1)$. The complexity of $\mathrm{Sum}(M, \bar{R}_0, \bar{R}_1)$ dominates this complexity; hence the aggregate complexity of the algorithm is $O(M E_{\max} Q^2 \bar{R}_0 \bar{R}_1)$.

V. Rounding-Based Complexity Scaling

Having simplified the globally optimal, strongly exponential $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ to the locally optimal DP-based weakly exponential $\mathrm{Sum}(i, R_0, R_1)$, we are now ready to perform the final step of our three-step optimization approach: we perform *rounding-based complexity scaling* to trade complexity for solution quality. By manipulating the DP tables used to store partially computed solutions, the two rounding techniques *DP dimension rounding* and *DP index rounding* reduce the number of table entries filled and, as a result, reduce complexity. We will discuss the two techniques in turn.

A. DP Dimension Rounding

The first rounding technique is *DP dimension rounding*. We first scale and round down overall budgets $\bar{R}_0$ and $\bar{R}_1$ by factor $K_{\mathrm{DR}} \in \mathcal{R}$—i.e., $\lfloor \bar{R}_0 / K_{\mathrm{DR}} \rfloor$ and $\lfloor \bar{R}_1 / K_{\mathrm{DR}} \rfloor$ —as input to the optimization. We then scale and round up costs of transmitting predicted frame $F_i, c(q_i) r_{i,j}$'s, by the same factor K_{DR}—i.e., $\lceil (c(q_i) r_{i,j}) / K_{\mathrm{DR}} \rceil$. Implementationally, we accordingly rewrite lines 8 and 12 of $\mathrm{Sum}(i, R_0, R_1)$ of Fig. 4 as

$$8.\ s := \mathrm{Sum}\left(i-1, R_0 - \left\lceil \frac{c(q0) r_{i,j}}{K_{\mathrm{DR}}} \right\rceil, R_1 - \left\lceil \frac{c(q1) r_{i,j}}{K_{\mathrm{DR}}} \right\rceil\right);$$

$$12.\ s := s + p(\mathbf{h}_i, q0, q1, r_{i,j}) * \mathrm{Prod}\left(j, i-1, R_0 - \left\lceil \frac{c(q0) r_{i,j}}{K_{\mathrm{DR}}} \right\rceil, R_1 - \left\lceil \frac{c(q1) r_{i,j}}{K_{\mathrm{DR}}} \right\rceil\right).$$

Similarly, we replace the cost terms in $\mathrm{Prod}(j, i, R_0, R_1)$ of Fig. 5 by rewriting lines 8 and 10 as

$$8.\ P := \mathrm{Prod}\left(j, i-1, R_0 - \left\lceil \frac{c(Y) r_{i,X}}{K_{\mathrm{DR}}} \right\rceil, R_1 - \left\lceil \frac{c(Z) r_{i,X}}{K_{\mathrm{DR}}} \right\rceil\right);$$

$$10.\ P := p(\mathbf{h}_i, Y, Z, r_{i,X}) * \mathrm{Prod}\left(X, i-1, R_0 - \left\lceil \frac{c(Y) r_{i,X}}{K_{\mathrm{DR}}} \right\rceil, R_1 - \left\lceil \frac{c(Z) r_{i,X}}{K_{\mathrm{DR}}} \right\rceil\right).$$

In so doing, instead of solving the original RQP selection instance I for locally optimal solution s^*, we solve an approximate instance I^A for solution s^A. Scaling down $\bar{R}_0$ and $\bar{R}_1$ means scaling down the dimension of the DP tables, hence the complexity is reduced by a factor of K_{DR}^2 at the cost of decreasing solution quality. Using $\mathrm{Sum}(i, R_0, R_1)$ and $\mathrm{Prod}(j, i, R_0, R_1)$ with the rewritten lines, the complexity of I^A is now $O(ME_{\max}Q^2\bar{R}_0\bar{R}_1K_{\mathrm{DR}}^{-2})$.

Note that, in the approximate instance I^A, the network resource constraints (9) become

$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}\left\lceil\frac{c(q0_i)r_{i,j}}{K_{\mathrm{DR}}}\right\rceil \le \left\lfloor\frac{\bar{R}_0}{K_{\mathrm{DR}}}\right\rfloor$$
$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}\left\lceil\frac{c(q1_i)r_{i,j}}{K_{\mathrm{DR}}}\right\rceil \le \left\lfloor\frac{\bar{R}_1}{K_{\mathrm{DR}}}\right\rfloor. \quad (12)$$

It is shown in Appendix C that solution s^A is feasible in I. Moreover, we can bound the performance difference between s^A and locally optimal s^* by first obtaining a super-optimal solution s^S in a new problem instance I^S, where we replace $\bar{R}_0$ and $\bar{R}_1$ with $\lceil\bar{R}_0/K_{\mathrm{DR}}\rceil$ and $\lceil\bar{R}_1/K_{\mathrm{DR}}\rceil$ and replace $c(q_i)r_{i,j}$'s with $\lfloor c(q_i)r_{i,j}/K_{\mathrm{DR}}\rfloor$. The super-optimal network resource constraints are

$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}\left\lfloor\frac{c(q0_i)r_{i,j}}{K_{\mathrm{DR}}}\right\rfloor \le \left\lceil\frac{\bar{R}_0}{K_{\mathrm{DR}}}\right\rceil$$
$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}\left\lfloor\frac{c(q1_i)r_{i,j}}{K_{\mathrm{DR}}}\right\rfloor \le \left\lceil\frac{\bar{R}_1}{K_{\mathrm{DR}}}\right\rceil. \quad (13)$$

After obtaining super-optimal solution s^S to I^S, we can bound our approximate solution s^A from the locally optimal s^* in the original problem instance I as follows:

$$\left|\mathrm{obj}(s^*) - \mathrm{obj}(s^A)\right| \le \left|\mathrm{obj}(s^S) - \mathrm{obj}(s^A)\right| \quad (14)$$

where $\mathrm{obj}(s)$ is the objective function (10) using solution s. The proof of performance bound (14) is also found in Appendix C.

B. DP Index Rounding

Instead of reducing the overall dimension of the DP table to scale down algorithmic complexity, another way is to limit the number of indexes used in the DP table given the table dimension. This rounding technique is called *DP index rounding*, and we accomplish that by always subtracting a positive integer multiple of $K_{\mathrm{IR}} \in \mathcal{I}$ from R_0 or R_1 during recursive calls in $\mathrm{Sum}(i, R_0, R_1)$ of Fig. 4. Implementationally, we do that by replacing $c(q_i)r_{i,j}$ with an approximate $K_{\mathrm{IR}}\lceil c(q_i)r_{i,j}/K_{\mathrm{IR}}\rceil$. Rewriting lines 8 and 12 of $\mathrm{Sum}(i, R_0, R_1)$, we obtain

$$8.\ s := \mathrm{Sum}\left(i-1, R_0 - K_{\mathrm{IR}}\left\lceil\frac{c(q0)r_{i,j}}{K_{\mathrm{IR}}}\right\rceil, R_1 - K_{\mathrm{IR}}\left\lceil\frac{c(q1)r_{i,j}}{K_{\mathrm{IR}}}\right\rceil\right)$$
$$10.\ s := s + p(\mathbf{h}_i, q0, q1, r_{i,j}) * \mathrm{Prod}\left(j, i-1, R_0 - K_{\mathrm{IR}}\left\lceil\frac{c(q0)r_{i,j}}{K_{\mathrm{IR}}}\right\rceil, R_1 - K_{\mathrm{IR}}\left\lceil\frac{c(q1)r_{i,j}}{K_{\mathrm{IR}}}\right\rceil\right).$$

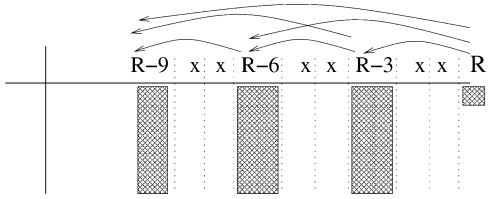


Fig. 6. Illustration of DP index rounding.

Similarly, we replace the cost terms in $\mathrm{Prod}(j, i, R_0, R_1)$ of Fig. 5 by rewriting lines 8 and 10 as

$$8.\ P := \mathrm{Prod}\left(j, i-1, R_0 - K_{\mathrm{IR}}\left\lceil\frac{c(Y)r_{i,X}}{K_{\mathrm{IR}}}\right\rceil, R_1 - K_{\mathrm{IR}}\left\lceil\frac{c(Z)r_{i,X}}{K_{\mathrm{IR}}}\right\rceil\right)$$
$$10.\ P := p(\mathbf{h}_i, Y, Z, r_{i,X}) * \mathrm{Prod}\left(X, i-1, R_0 - K_{\mathrm{IR}}\left\lceil\frac{c(Y)r_{i,X}}{K_{\mathrm{IR}}}\right\rceil, R_1 - K_{\mathrm{IR}}\left\lceil\frac{c(Y)r_{i,X}}{K_{\mathrm{IR}}}\right\rceil\right).$$

As an example, we see an illustration of DP index rounding in Fig. 6 when $K_{\mathrm{IR}} = 3$. By recursing only on R less multiples of 3, we are only filling at most 1/3 of all indexes along both the R_0 and R_1 dimensions. Hence, the new algorithmic complexity is $O(ME_{\max}Q^2\bar{R}_0\bar{R}_1K_{\mathrm{IR}}^{-2})$.

The new network constraints using DP index rounding are as follows:

$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}K_{\mathrm{IR}}\left\lceil\frac{c(q0_i)r_{i,j}}{K_{\mathrm{IR}}}\right\rceil \le \bar{R}_0$$
$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}K_{\mathrm{IR}}\left\lceil\frac{c(q1_i)r_{i,j}}{K_{\mathrm{IR}}}\right\rceil \le \bar{R}_1. \quad (15)$$

Using a similar opposite rounding technique in the previous section, we can bound the performance of the approximate solution s^A from the locally optimal solution s^* by first constructing a super-optimal solution s^S and evaluating bound (14). The proof will be similar to that in the DP dimension rounding case (shown in the Appendix C) and hence is omitted here.

C. Applying DP Dimension and Index Rounding

We can employ both rounding strategies simultaneously: replace $\bar{R}_0$ and $\bar{R}_1$ with $\lfloor\bar{R}_0/K_{\mathrm{DR}}\rfloor$ and $\lfloor\bar{R}_1/K_{\mathrm{DR}}\rfloor$, respectively, as input to the algorithm; and replace $c(q)r_{i,j}$ with $K_{\mathrm{IR}}\lceil c(q)r_{i,j}/(K_{\mathrm{IR}}K_{\mathrm{DR}})\rceil$ in lines 8 and 12 of recursive function $\mathrm{Sum}(i, R_0, R_1)$ of Fig. 4 and lines 8 and 10 of $\mathrm{Prod}(j, i, R_0, R_1)$ of Fig. 5. The resulting network constraints are

$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}K_{\mathrm{IR}}\left\lceil\frac{c(q_i)r_{i,j}}{K_{\mathrm{IR}}K_{\mathrm{DR}}}\right\rceil \le \left\lfloor\frac{\bar{R}_0}{K_{\mathrm{DR}}}\right\rfloor$$
$$\sum_{i=1}^{M}\sum_{\forall j|e_{i,j}\in\mathcal{E}} x_{i,j}K_{\mathrm{IR}}\left\lceil\frac{c(q_i)r_{i,j}}{K_{\mathrm{IR}}K_{\mathrm{DR}}}\right\rceil \le \left\lfloor\frac{\bar{R}_1}{K_{\mathrm{DR}}}\right\rfloor. \quad (16)$$

The resulting complexity is $O(ME_{\max}Q^2\bar{R}_0\bar{R}_1K_{\rm DR}^{-2}K_{\rm IR}^{-2})$. An interesting question is then: given a desired complexity reduction factor K^2, where $K = K_{\rm DR}K_{\rm IR}$, what are the tradeoffs in using different $K_{\rm DR}$ and $K_{\rm IR}$?

Because our approximation bound (14) is an *a posteriori* bound instead of an *a priori* one, i.e., we do not know precisely the extent of the error until approximate solution s^A and super-optimal solution s^S are computed and evaluated, we cannot directly relate the performance of our approximate solution s^A to $K_{\rm DR}$ and $K_{\rm IR}$ analytically. To estimate the performance of the to-be-constructed approximate solution s^A *a priori* given rounding factors $K_{\rm DR}$ and $K_{\rm IR}$, we instead focus on an alternate performance metric $\Omega_{\rm err}$ that tracts the maximum possible rounding error to occur when calculating network resource constraints (16) instead of the original (9). In the worst case, $\Omega_{\rm err}$ is the maximum rounding error on the right-hand side of (16) plus the maximum rounding error on the left-hand side. Right maximum error is the maximum rounding error between $\bar{R}_0$ and $\bar{R}_1$; left maximum error is the number of P-frames $(M-1)$ times the maximum rounding error of $c(q)r_{i,j}$:

$$\begin{aligned}\Omega_{\rm err} &= \max\left\{\left|\bar{R}_0 - K_{\rm DR}\left\lfloor\frac{\bar{R}_0}{K_{\rm DR}}\right\rfloor\right|, \left|\bar{R}_1 - K_{\rm DR}\left\lfloor\frac{\bar{R}_1}{K_{\rm DR}}\right\rfloor\right|\right\} \\ &\quad + (M-1) * \max_{q_i, r_{i,j}}\left|c(q_i)r_{i,j} - K_{\rm IR}K_{\rm DR}\left\lceil\frac{c(q_i)r_{i,j}}{K_{\rm IR}K_{\rm DR}}\right\rceil\right| \\ &= K_{\rm DR} + (M-1)K_{\rm IR}K_{\rm DR}. \end{aligned} \tag{17}$$

If we now substitute $K_{\rm IR} = K/K_{\rm DR}$ into (17), we obtain

$$\Omega_{\rm err} = K_{\rm DR} + (M-1)K. \tag{18}$$

Hence, $\Omega_{\rm err}$ is a linear increasing function of $K_{\rm DR}$, i.e., we should let $K_{\rm IR} = K$ to minimize $\Omega_{\rm err}$ for fixed K. Depending on implementation, in practice, we may need to use a larger $K_{\rm DR}$ to reduce the amount of memory needed for the DP tables, each of dimension $O(ME_{\max}Q^2\bar{R}_0\bar{R}_1K_{\rm DR}^{-2})$. Thus, a practical rounding factor selection strategy to achieve a complexity scaling factor of $K^2, K = K_{\rm DR}K_{\rm IR}$, is as follows.

1) Select the smallest $K_{\rm DR} \in \mathcal{R}$ that sufficient memory can be allocated for DP tables.
2) Given K and $K_{\rm DR}$, calculate $K_{\rm IR} := \lceil K/K_{\rm DR}\rceil$.

VI. Experimentation

A. Numerical Comparison of Optimal and Locally Optimal Algorithms

In this experimental section, we begin with a numeric comparison between the globally optimal algorithm $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ and the locally optimal algorithm $\mathrm{Sum}(i, R_0, R_1)$. For network QoS, we assume simple multiple transmissions where $q0_i(q1_i)$ means $q0_i(q1_i)$ transmissions on path 0 (1). Accordingly, the cost vector is simply $c(q) = q$. We performed two trials, with raw path loss rates at $(\alpha_0, \alpha_1) = (0.10, 0.06)$ and $(\alpha_0, \alpha_1) = (0.08, 0.04)$, respectively. The total bandwidth of both paths are kept constant while the bandwidth of path 1 is varied.

For application-level inputs to the optimization—encoding rates $r_{i,j}$'s, we use H.264 version JM8.4 [18] to encode two 300-frame QCIF (176×144) sequences subsampled in time by 2: MPEG test sequence `news` and `mother`. For `news`, we held quantization parameters at 25 and 20 for I-frames and P-frames, respectively, resulting in source coding rate 140.38 kbps if each P-frame F_i is coded using its previous frame F_{i-1}. For `mother`, we held quantization parameters at 22 and 18 for I-frames and P-frames, respectively, resulting in source coding rate 137.59 kbps if each P-frame F_i is coded using F_{i-1}.

To get rates $r_{i,j}$'s, we iteratively force each predicted frame F_i to use reference frame F_{i-t} for motion prediction during iteration $t = \{0, 1, 2, 3, 4, 5\}$. The resulting coding rate is $r_{i,i-t}$. We assume a predicted frame F_i will use a reference frame no further back in time than F_{i-5}, or simply $E_{\max} = 5$.

For this part of the experiment only, we optimized only the first seven frames for each data point. The relative small number of frames (7) being optimized is selected because optimal $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ is exponential in the number of frames. The rounding parameters of the locally optimal algorithm are set at $K_{\rm IR} = 1$ and $K_{\rm DR} = 100$. We let the total available bandwidth for both sequences be 150 kbps for the two trials, which roughly corresponds to a 10% overhead for loss protection beyond source coding. The objective function for globally optimal $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ and the locally optimal $\mathrm{Sum}(i, R_0, R_1)$ is shown in Fig. 7. We see that the performance of globally and locally optimal curves are very similar for both trials 1 and 2; the largest relative difference is only 3.79% and 3.07% for `news` and 0.83% and 1.10% for `mother` for the two trials. We can therefore safely conclude that the developed locally optimal algorithm is sufficient as a starting point for later algorithmic development.

An interesting observation in Fig. 7 is that, while all performance curves have overall upward movement—this is expected since path 1 has a lower packet loss rate in both trials—the curves dip before moving to a higher plateau, resulting in nonmonotonicity of the performance curves. The reason is that the optimization is formulated as a discrete resource allocation problem: a frame in path 0 will be reassigned to path 1 when sufficient path 0 bandwidth has been reallocated to path 1, but will not fit in either path when only small increments are shifted from one path to the other, resulting in lower performance. This nonmonotonicity of performance curves will be a recurring characteristic in later experiments as well.

B. Experiment Setup

To test the rounding-based complexity scaling algorithm in a network simulated environment, we developed a network simulator called (mu)ltiple-path (n)etwork (s)imulator (`muns`), shown in Fig. 8, that was also used in other network experiments [19]. Each transmission path k is implemented as a queue of constant service rate μ_k, followed by an independent and identically distributed (iid) packet erasure channel with shifted Gamma distributed delay. Upon each packet arrival, the client informs the server of its status using ACK with client feedback delay $D_F = 0$. Queue service rates μ_k's are set to create bottlenecks if links are utilized more than their preassigned bandwidths. In our experiment, the volume of packets in each queue (path) is controlled at the application, and thus overutilization does not happen. As such, μ_k's are not

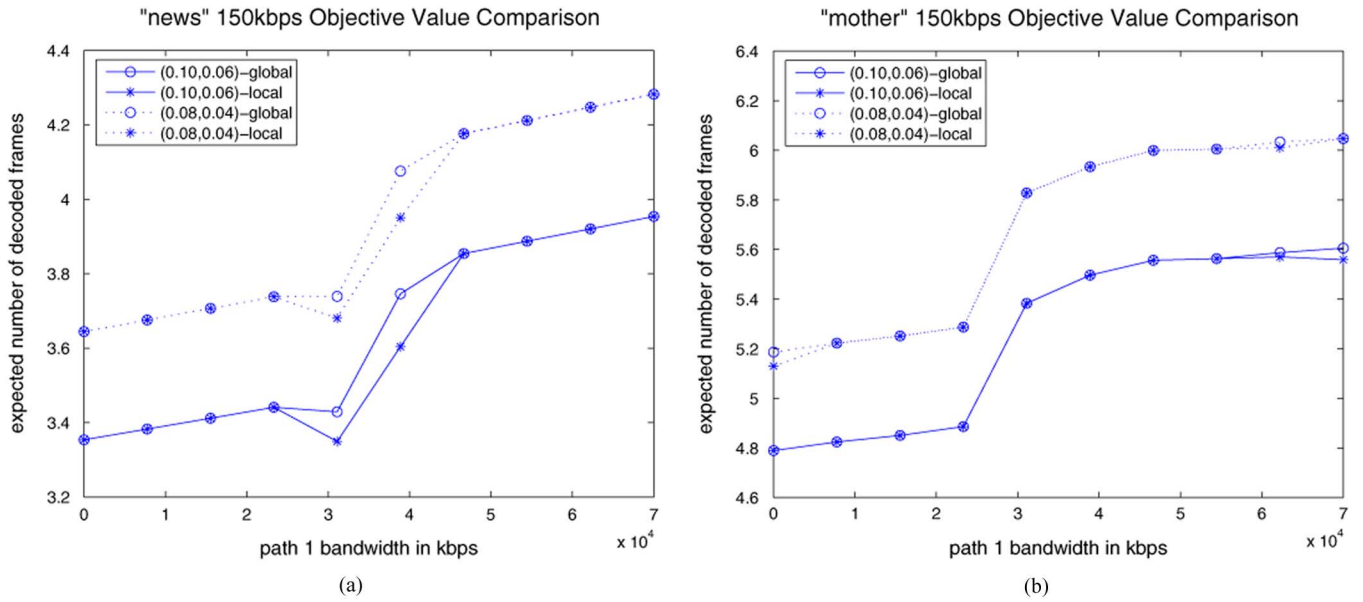


Fig. 7. Numerical comparison for `news` and `mother` for varying path–1 bandwidth for fixed total $\mathrm{bandwidth} = 150$ kbps. (a) Objective value for `news`. (b) Objective value for `mother`.

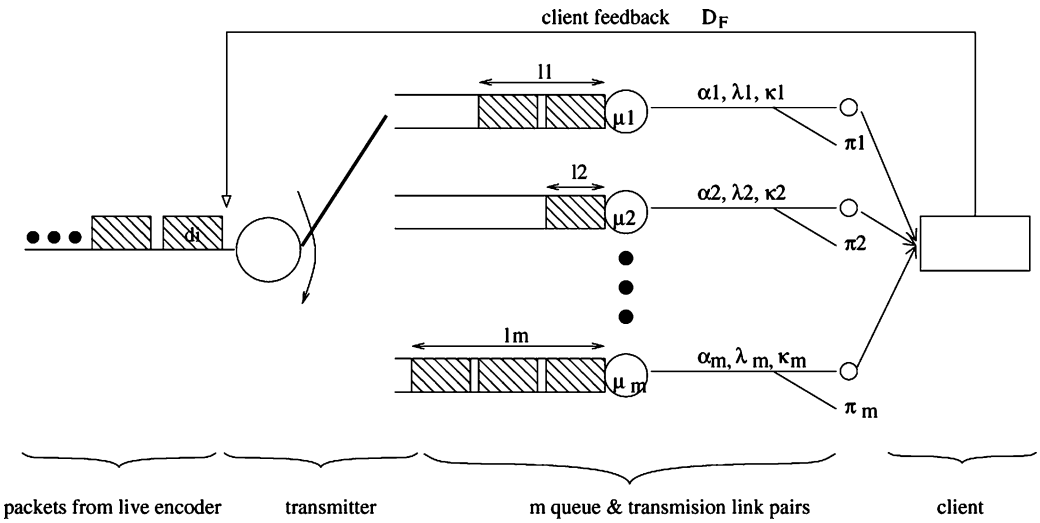


Fig. 8. Network simulator `muns` is used to simulate queuing, transmission, and losses of packets.

necessary, and we set each to 0 ms. The optimization period P is set to 300 ms, the optimization window size is $M = 10$, and the network MTU is $\mathrm{MTU} = 1500$ bytes.

For the application-level inputs, we use the same encoding rates $r_{i,j}$'s as in Section VI-A. The shifted Gamma distribution parameters for network delay used in the experiments are $(\alpha_0, \lambda_0, \kappa_0) = (4, 0.1, 60\,\mathrm{ms})$ and $(\alpha_1, \lambda_1, \kappa_1) = (3, 0.1, 60\,\mathrm{ms})$ for the two transmission paths. The delay mean and variance are 100 ms and 400 ms^2 and 90 ms and 300 ms^2, for the two paths, respectively.

C. Experimental Results 1: RQP Selection Comparison

For the first set of simulation experiments, we show that our optimization, as a streaming optimization scheme, has practical merits and outperforms two competing ad hoc schemes. For both sequences `news` and `mother`, we first fixed the combined bandwidth of the two paths $\bar{R}_0 + \bar{R}_1$ at 150 kps as done previously. Rounding parameters K_{DR} and K_{IR} were held constant at 1000 and 1, respectively. By varying the share of 150-kbps bandwidth allocated to the second path $\bar{R}_1$, we tracked the corresponding performance at the client in PSNR. PSNR was calculated as follows. First, a frame F_i was deemed correctly decoded if and only if F_i was timely and correctly delivered *and* all its dependent frames were correctly decoded. If a frame F_i was correctly decoded, then PSNR for F_i was computed using the decoded F_i and the original uncompressed F_i. If a frame F_i was not decodable, the most recent correctly decoded frame F_j was used as a replacement, and the PSNR for F_i was computed using the decoded F_j and original uncompressed F_i. The sequence was replayed 300 times for an averaging effect. Two trials of different packet loss rates of the two paths were performed: (0.10,0.06) and (0.08,0.04).

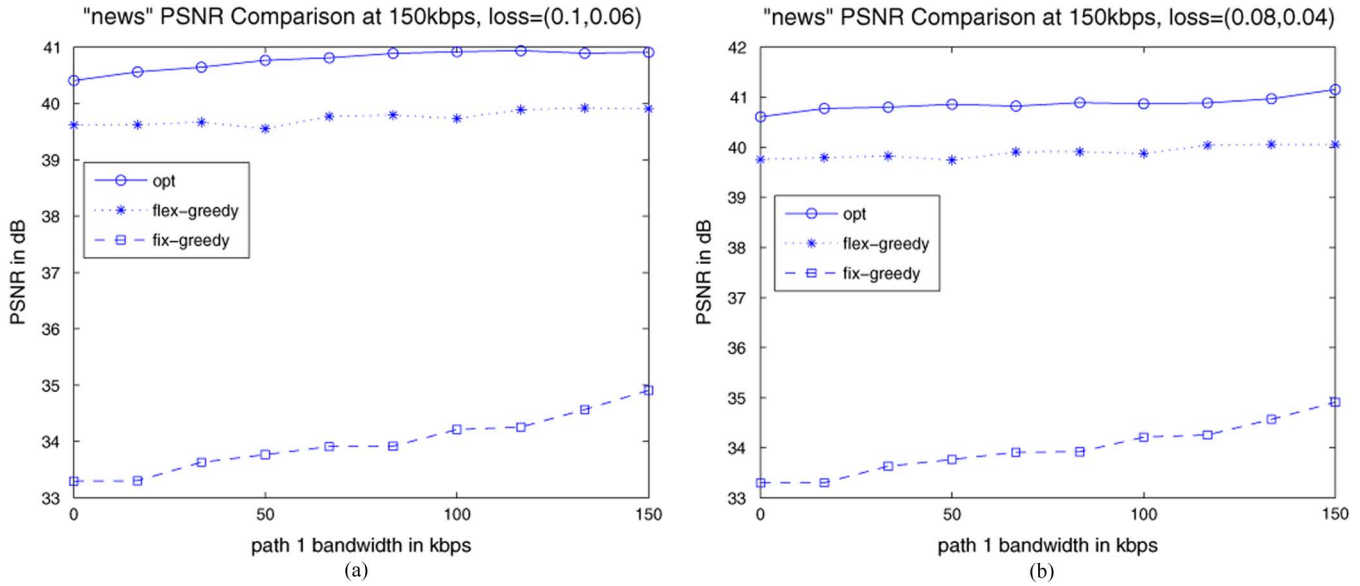


Fig. 9. Streaming performance for news in PSNR for varying path–1 bandwidth for fixed total bandwidth = 150 kbps. (a) loss rate = $(0.10, 0.06)$. (b) loss rate = $(0.08, 0.04)$.

We compare our locally optimal algorithm opt to two greedy selection schemes we call fix-greedy and flex-greedy that contain elements of the simplified version of the RaDiO framework[3] [2]. The fix-greedy scheme works as follows. First, fix-greedy assumes a fixed differential coding structure where an I-frame is inserted every ten frames and other frames F_i's motion-compensate from previous frame F_{i-1}'s. Given budgets $\bar{R}_0$ and $\bar{R}_1$ in the two paths, fix-greedy then incrementally selects a frame in the optimization window with the best combination of QoS and delivery path that maximizes the benefit-to-cost ratio, where the benefit is the increase in objective value (10), and the cost is the increase in bit expenditure. fix-greedy proceeds with the selection until both budgets are expended. flex-greedy operates in a similar fashion as fix-greedy, with the additional flexibility of selecting a reference frame from set $\{F_i, \ldots, F_{i-5}\}$ greedily for each frame F_i in the optimization window. To the best of the authors' knowledge, fix-greedy and flex-greedy represent the best performing complexity-efficient selection algorithms available in the literature.

The performance for the sequence news in PSNR, as a function of the first path bandwidth $\bar{R}_0$, is shown in Fig. 9(a) and 9(b) for the two trials, respectively. In both trials, we see that the PSNR increases as $\bar{R}_0$ increases. This is expected since path 1 has a lower loss rate than path 0. Further, we see that opt outperformed fix-greedy in PSNR by up to 7.25 dB and outperformed flex-greedy by up to 1.21 dB for the first trial, and outperformed fix-greedy by up to 7.47 dB and outperformed flex-greedy by up to 1.11 dB for the second trial. The large performance difference indicates the effectiveness of opt for the RQP problem.

Under the same test conditions, we next generated the performance plots for sequence mother shown in Fig. 10(a) and (b) for the two trials. We see similar trends as we saw previously for sequence news. Specifically, we see that opt outperformed fix-greedy in PSNR by up to 6.16 dB and outperformed flex-greedy by up to 0.66 dB for the first trial and outperformed fix-greedy by up to 5.39 dB and outperformed flex-greedy by up to 0.73 dB for the second trial. The noticeable performance difference again indicates the effectiveness of opt for the RQP problem.

[3]RaDiO is a packet scheduling algorithm that does not alter the encoding of the source. We are merely extending the idea of the simplified RaDiO— greedily selecting the most beneficial grouping per bit overhead—to the RQP problem.

D. Experimental Results 2: Performance/Complexity Tradeoff Using Dimension Rounding

We have already discussed how complexity reduction can be achieved by varying rounding parameters. In practice, we desire the objective function, such as the number of correctly decoded frames, to change *gradually* as we vary the rounding parameter. Clearly, if there is a drastic drop in objective function when the rounding parameter is larger than a certain small value, then any rounding parameter larger than that small value is not likely to be chosen in practice, resulting in a very limited range of useful complexity scaling. In the next experiment, we examine the change in objective function as the rounding parameter is gradually increased to examine the range of useful complexity scaling.

We held the bandwidth of the two paths $\bar{R}_0$ and $\bar{R}_1$ constant at (50 kps, 100 kps) for sequence news and mother. DP index rounding parameter K_{IR} was kept constant at 1, and K_{DR} was varied to observe the tradeoff between performance and complexity; recall the complexity of the optimization is $O(ME_{\max}Q^2\bar{R}_0\bar{R}_1K_{\mathrm{DR}}^{-2}K_{\mathrm{IR}}^{-2})$. Again, we performed two trials of different packet loss rates of the two paths: (0.10,0.06) and (0.08,0.04). The performance in PSNR as a function of K_{DR} for both trials can be seen in Fig. 11(a) for sequence news and in Fig. 11(b) for sequence mother.

We see in both Fig. 11(a) and (b) that, indeed, as DP dimension rounding factor K_{DR} increased, the quality of the solution suffered due to rounding, and the performance decreased for

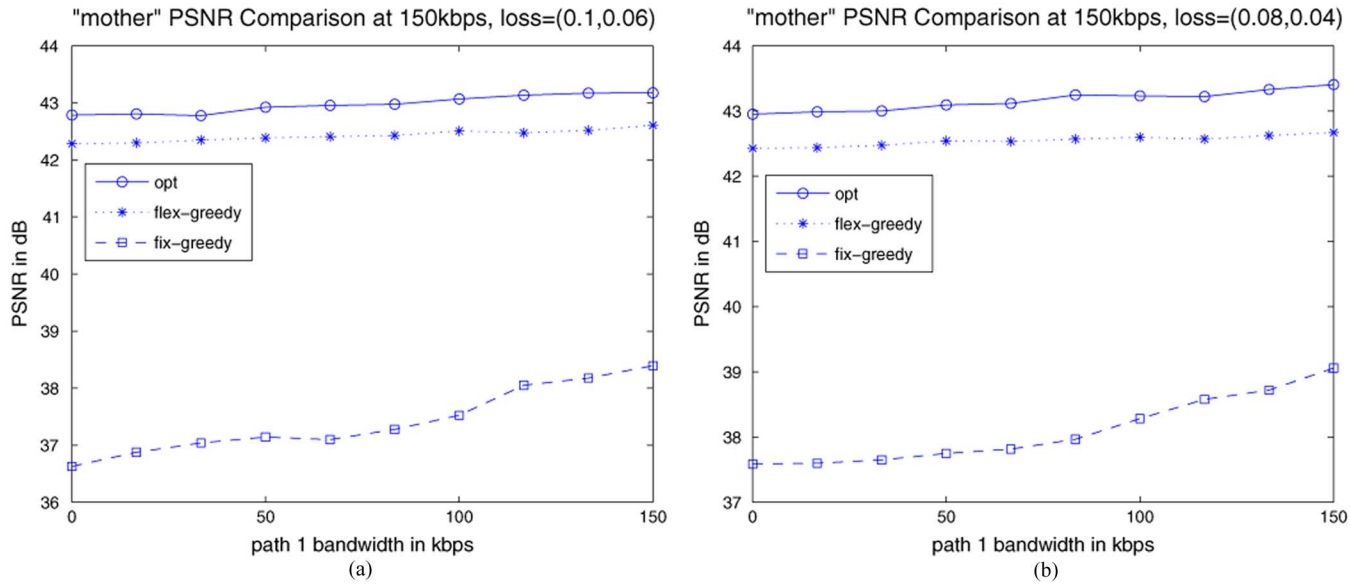


Fig. 10. Streaming performance for mother in PSNR for varying path–1 bandwidth for fixed total $\mathrm{bandwidth} = 150$ kbps. (a) $\mathrm{loss\ rate} = (0.10, 0.06)$. (b) $\mathrm{loss\ rate} = (0.08, 0.04)$

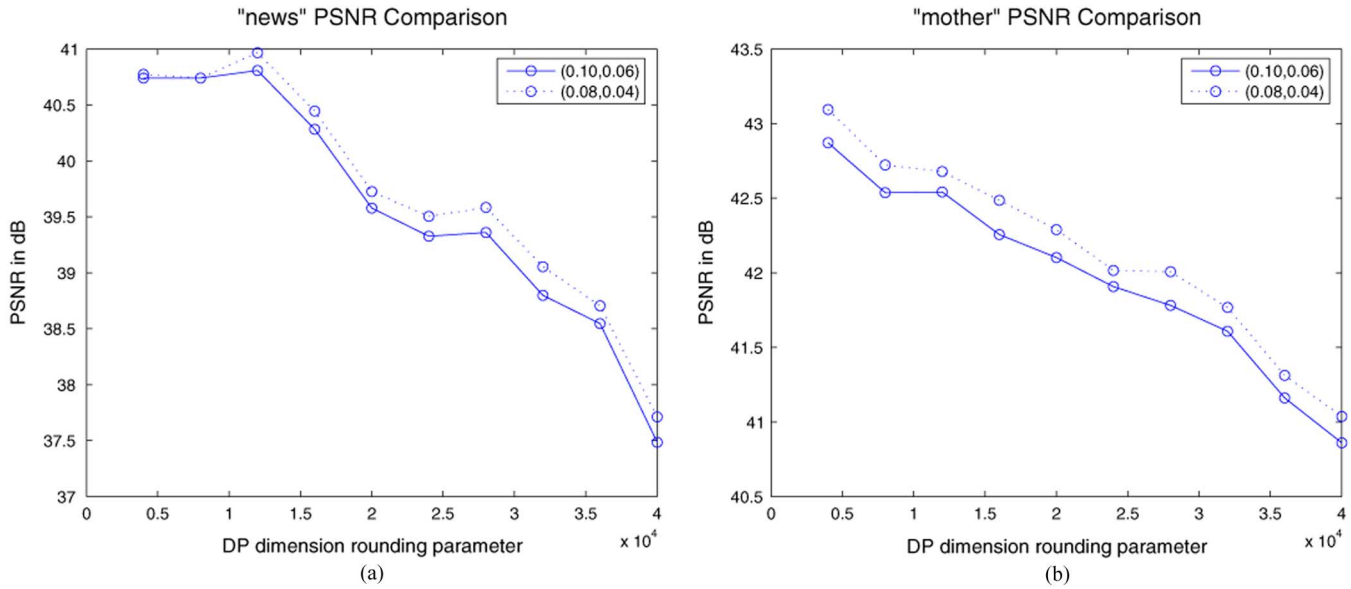


Fig. 11. Visual quality degradation of news and mother as dimension rounding parameter (K_{DR}) increases. (a) PSNR for news. (b) PSNR for mother.

both trials and for both sequences. More importantly, we see that the approximation error, as indicated by the degradation in PSNR, decreases gradually over a wide range of rounding parameters. This suggests that a very wide range of useful complexity scaling can be realized using the dimension rounding parameter K_{DR}. We also observed that PSNR does not decrease monotonically with increasing rounding parameter. This can be partially attributed to the fact that rounding is a nonlinear operation, meaning that the precise degree of the rounding error will depend on actual numbers $r_{i,j}$'s, $\bar{R}_0$, and $\bar{R}_1$ as well as K_{DR}. The general downward trend of the curves, however, is in agreement with our analysis in Section V-A that performance is in general inversely proportional to rounding factor K_{DR}.

E. Experimental Results 3: Performance/Complexity Tradeoff Using Index Rounding

In the third experiment, we show that a similar performance/complexity tradeoff can be accomplished using the index rounding parameter K_{IR} instead of the dimension rounding parameter K_{DR}. As in the second experiment, the bandwidths of the two paths for news were held constant at (50 kps, 100 kps). Packet loss rates for the two paths were again held at (0.10, 0.06) and (0.08, 0.04), respectively, for two trials. This time we held K_{DR} constant at 1000 as K_{IR} is varied. Quality degradation in PSNR as a function of index parameter K_{IR} is shown in Fig. 12(a) for sequence news and in Fig. 12(b) for sequence mother.

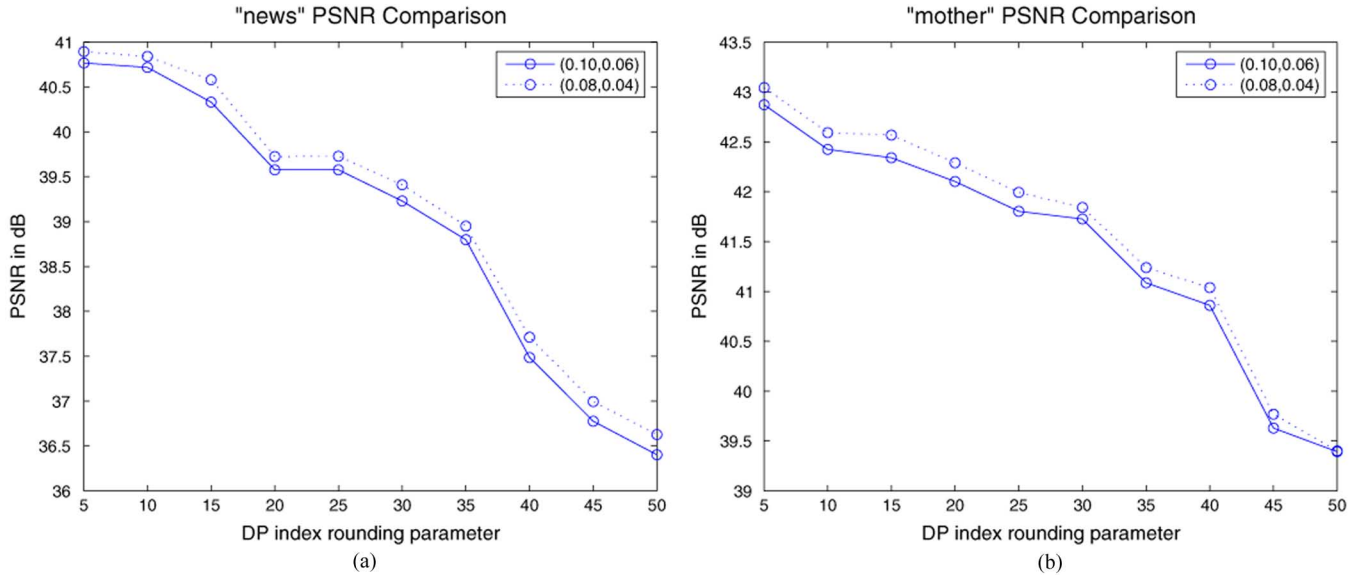


Fig. 12. Visual quality degradation of news and mother as index rounding parameter (K_{IR}) increases. (a) PSNR for news. (b) PSNR for mother.

We first observed in Fig. 12 that the general downward trend and nonmonotonicity of the curves are similar to those in Fig. 11. This is expected since the characteristics of the gradually increasing round-off error and the nonlinearity of the rounding operation remain. We also observed that the curves did *gracefully* degrade over most range of K_{IR}. This is perhaps surprising, since at $K_{\mathrm{IR}} = 30$, we have very large computation reduction factor $K = K_{\mathrm{DR}}K_{\mathrm{IR}}$ of 30 000. This means that, using K_{DR} and K_{IR}, a very large useful range of performance/complexity tradeoff can be employed in practice.

VII. Conclusion

In this paper, we studied the optimization problem of jointly selecting reference frames for motion prediction, and the path and QoS level for transport in a multipath streaming scenario. In particular, we presented a low-complexity approximate optimization scheme that produces results comparable to the optimal. We also presented and evaluated several rounding techniques to allow multiple complexity–quality tradeoff points for the approximate optimization scheme. Our approach is novel in that, unlike conventional Lagrangian approaches, it uses a mixture of two rounding techniques, DP dimension rounding and DP index rounding, to gradually trade complexity for quality of the obtained solution. Experiments using H.264 showed that graceful complexity–quality tradeoff can be achieved over a wide range.

Appendix I
NP-Hard Proof of RQP Selection Problem

Here, we prove that the RQP selection problem is NP-hard by proving that the corresponding binary decision problem—does there exist a solution such that the objective value is larger than some constant C—is NP-complete. We accomplish that, via a reduction from a well-known NP-complete problem, *Knapsack* problem [17, p. 247]). For completeness sake, the Knapsack problem is repeated from [17] here.

INSTANCE: Finite set $\mathcal{U}$, for each $u \in \mathcal{U}$ a size $s(u) \in Z^+$ and a value $v(u) \in Z^+$, and positive integers B and C.
QUESTION: Is there a subset $\mathcal{U}' \subseteq \mathcal{U}$ such that $\sum_{u\in\mathcal{U}'} s(u) \leq B$ and $\sum_{u\in\mathcal{U}'} v(u) \geq C$?

The problem remains NP-complete if $v(u) = s(u)$.

For the reduction, we construct a corresponding RQP selection problem instance as follows. We construct a $|\mathcal{U}|+1$-frame sequence, each frame $F_i, i > 1$, having one possible RF, which is F_1. Each frame F_i has a rate $r_{i,1} = s(u_{i-1})$. We construct the QoS set to offer only two services: $\mathcal{Q} = \{0, 1\}$. The resulting rate matrix $\mathbf{r}$ is

$$\mathbf{r} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ s(u_1) & 0 & \dots & 0 \\ s(u_2) & 0 & \dots & 0 \\ \vdots & & & \\ s(u_{|\mathcal{U}|}) & 0 & \dots & 0 \end{bmatrix}. \tag{19}$$

There is only one path with sufficient bandwidth budget for packet transmission; i.e., $\bar{R}_1 < \min_{i\in\mathcal{U}} s(u_i)$. The corresponding construction for a five-frame subsequence is shown in Fig. 13. The resulting RQP selection problem under this construction mathematically becomes

$$\max_{\{x_{i,1}\}} \left\{ \sum_{i=2}^{|\mathcal{V}|} x_{i,1} \frac{r_{i,1}}{B} \right\} \quad \text{s.t.} \sum_{i=2}^{|\mathcal{V}|} x_{i,1} r_{i,1} \leq B \tag{20}$$

where we set $p(\mathbf{h}_1, 1, r_{i,1})$ and $\bar{R}_0$ in (10) to be $r_{i,1}/B$ and B, respectively. The corresponding binary decision problem is: does there exist a RQP selection—$x_{i,1} \in \{0, 1\}$—such that the objective value is ≥ 1?

It is clear from (20) that the binary decision problem of the constructed RQP selection problem is equivalent to the original Knapsack problem instance when $s(u) = v(u)$. Hence, the RQP selection problem is as least as hard as the Knapsack problem. Therefore, the RQP selection problem is NP-hard.

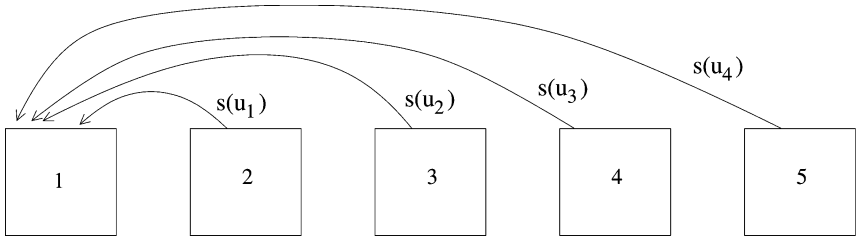


Fig. 13. Joint selection selection of RQP for this graph can be shown to be equivalent to the *Knapsack* problem, which is NP-complete.

APPENDIX II
PROOF OF OPTIMALITY OF GLOBALLY OPTIMAL ALGORITHM

We now prove that $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$, described in Fig. 3 of Section IV, is indeed globally optimal. More precisely, we want to show that $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ returns the maximum expected number of frames for the i-frame sub-sequence $\{F_1, \ldots, F_i\}$ *with benefits*, given resources R_0 and R_1 in paths 0 and 1 are available to the subsequence. By benefits, designated by benefit vector $\mathbf{w}$ of length M, we mean that, for $1 \leq j \leq i$, the successful decoding of F_j brings in *dependent benefit* w_j as well, and for $i+1 \leq j \leq M$, there is *independent benefit* w_j.

We prove by induction. For the base case, we show that $\mathrm{Sum}(1, R_0, R_1, \mathbf{w})$ is optimal. Since F_1 is the only frame under consideration and it has no previous frame to reference, by searching through all possible QoS levels $q0$ and $q1$ for F_1 without exceeding the resource budgets R_0 and R_1 (line 4 of Fig. 3), and subsequently adding the independent benefits $w_j, 2 \leq j \leq M$ (line 5), we can find the optimal solution.

For the inductive case, by assuming $\mathrm{Sum}(i, R_0', R_1', \mathbf{w}')$ is optimal, we show $\mathrm{Sum}(i+1, R_0, R_1, \mathbf{w})$ is optimal. We consider two subcases. Suppose F_{i+1} is best selected as a P-frame. Then the successful decoding of its chosen reference frame F_j would mean F_{i+1} can successfully decode with probability $p(\mathbf{h}_{i+1}, q0, q1, r_{i+1,j})$. In turn, the successful decoding of F_{i+1} would mean we get one more correctly decoded frame (F_{i+1}) *plus* F_{i+1}'s dependent benefit w_{i+1}. Given F_{i+1}'s selection, the problem is then equivalent to adding the contribution of $F_{i+1}, p(\mathbf{h}_{i+1}, q0, q1, r_{i+1,j})(1 + w_{i+1})$, into a dependent benefit for F_j (line 12), eliminating any independent benefit for F_{i+1} (line 13), and solving the reduced i-frame problem $\mathrm{Sum}(i, R_0 - c(q0)r_{i+1,j}, R_1 - c(q1)r_{i+1,j}, \mathbf{w}')$ with new benefit vector $\mathbf{w}'$ (line 16).

Now suppose F_{i+1} is best selected as an I-frame. Then the successful decoding of F_{i+1} depends on no other frames, and from its own successful delivery we reap one more correctly decoded frame (F_{i+1}) and its benefit w_{i+1}. Given F_{i+1}'s selection as an I-frame, the problem is then equivalent to setting F_{i+1}'s independent benefit to $p(\mathbf{h}_{i+1}, q0, q1, r_{i+1,j})(1+w_{i+1})$ (line 15) and solving the reduced i-frame problem $\mathrm{Sum}(i, R_0 - c(q0)r_{i+1,j}, R_1 - c(q1)r_{i+1,j}, \mathbf{w}')$ with new benefit vector $\mathbf{w}'$ (line 16).

Since we assumed earlier that $\mathrm{Sum}(i, R_0', R_1', \mathbf{w}')$ is itself optimal, by searching through all possible choices for F_{i+1}—reference frames and QoS levels, $\mathrm{Sum}(i+1, R_0, R_1, \mathbf{w})$ must necessarily return the optimal solution. Since both the base and inductive cases are proven, we also proved $\mathrm{Sum}(i, R_0, R_1, \mathbf{w})$ is optimal. Finally, since initially no frames F_i's in the M-frame subsequence has any benefits, $\mathrm{Sum}(M, R_0, R_1, \mathbf{0})$ returns the optimal expected number of correctly decoded frames as claimed.

APPENDIX III
PROOF OF ROUNDING-BASED COMPLEXITY SCALING

To prove feasibility of approximate solution s^A in I and the performance bound (14), we essentially need to prove two axioms: 1) that s^A satisfies original network constraints (9) and 2) that s satisfies super-optimal network constraints (13) in I^S. To prove the first axiom, we first let the approximate solution be $s^A = (\{x_{i,j}^A\}, \{q_i^A\}, \{t_i^A\})$. Given the s^A satisfies the first network constraint in (12), we can write

$$\sum_{i=1}^{M} \sum_{\forall j | e_{i,j} \in \mathcal{E}} x_{i,j}^A \left\lceil \frac{c\left(q0_i^A\right) r_{i,j}}{K_{\mathrm{DR}}} \right\rceil K_{\mathrm{DR}} \leq \left\lfloor \frac{\bar{R}_0}{K_{\mathrm{DR}}} \right\rfloor K_{\mathrm{DR}}$$

$$\sum_{i=1}^{M} \sum_{\forall j | e_{i,j} \in \mathcal{E}} x_{i,j}^A \frac{c\left(q0_i^A\right) r_{i,j}}{K_{\mathrm{DR}}} K_{\mathrm{DR}} \leq \frac{\bar{R}_0}{K_{\mathrm{DR}}} K_{\mathrm{DR}} \quad (21)$$

where (21) holds since $(c(q0_i)r_{i,j})/(K_{\mathrm{DR}}) \leq \lceil (c(q0_i)r_{i,j})/(K_{\mathrm{DR}}) \rceil$ and $\bar{R}_0/K_{\mathrm{DR}} \geq \lfloor \bar{R}_0/K_{\mathrm{DR}} \rfloor$. Similar steps can be done for the second network constraint. Therefore, the first axiom holds and s^A is feasible in I. Using a similar argument, one can show easily the second axiom: that s is feasible in I^S. By local optimality of s^S in solution space of I^S, we have

$$\mathrm{obj}(s^S) \geq \mathrm{obj}(s). \quad (22)$$

By subtracting $\mathrm{obj}(s^A)$ and taking absolute value on both sides, we get (14).

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

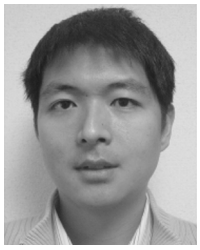
[2] P. Chou and Z. Miao, Rate-distortion Optimized Streaming of Packetized Media Microsoft Research, Tech. Rep. MSR-TR-2001-35, Feb. 2001.

[3] T. Wiegand, N. Farber, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE J. Sel. Areas. Commun.*, vol. 18, no. 6, pp. 1050–1062, Jun. 2000.

[4] Y. Liang, M. Flierl, and B. Girod, "Low-latency video transmission over lossy packet networks using rate-distortion optimized reference picture selection," in *Proc. IEEE Int. Conf. Image Process.*, Rochester, NY, Sep. 2002, pp. II-181–II-184.

[5] C.-M. Huang, K.-C. Yang, and J.-S. Wang, "Error resilience supporting bi-directional frame recovery for video streaming," in *Proc. IEEE Int. Conf. Image Process.*, Singapore, Oct. 2004, pp. 537–540.

[6] J. Apostolopoulos, "Error-resilient video compression via multiple state streams," in *Proc. Int. Workshop Very Low Bitrate Video Coding (VLBV'99)*, Oct. 1999, pp. 168–171.
[7] Y. Liang, E. Setton, and B. Girod, "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in *Proc. IEEE Workshop Multimedia Signal Processing*, St. Thomas, U.S. Virgin Islands, Dec. 2002.
[8] O. H. Ibarra and C. E. Kim, "Fast approximation algorithms for the knapsack and sum of subset problems," *J. ACM*, vol. 22, pp. 463–468, 1975.
[9] V. Vazarini, *Approximation Algorithms*. Berlin, Germany: Springer-Verlag, 2001.
[10] G. Cheung and C. Chan, "Jointly optimal reference frame & quality of service selection for H.26L video coding over lossy networks," in *Proc. IEEE Int. Conf. Multimedia Expo*, Baltimore, MD, Jul. 2003, pp. II-49–II-52.
[11] G. Cheung, "Near-optimal multipath streaming of h.264 using reference frame selection," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, Sep. 2003, pp. III-653–III-656.
[12] *3GPP TS 26.233 Transparent End-to-End Packet Switched Streaming Services (PSS); General description (Release 4)*, , Mar. 2001 [Online]. Available: ftp://ftp.3gpp.org/Specs/2001-03/Rel-4/26_series/26233-400.zip
[13] *3GPP TS 26.234 Transparent End-to-End Packet Switched Streaming Services (PSS); Protocols and codecs (Release 4)*, , Mar. 2001 [Online]. Available: ftp://ftp.3gpp.org/Specs/2001-03/Rel-4/26_series/26234-400.zip
[14] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 43–56.
[15] Y. Liang, J. Apostolopoulos, and B. Girod, "Model-based delay-distortion optimization for video streaming using packet interleaving," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2002, pp. 1315–I319.
[16] Y. Liang, J. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: Does burst length matter?," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Hong Kong, Apr. 2003, pp. V-684–V-687.
[17] M. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
[18] *The TML Project Web-Page and Archive*, [Online]. Available: http://kbc.cs.tu-berlin.de/stewe/vceg/
[19] G. Cheung, P. Sharma, and S. J. Lee, "Striping delay-sensitive packets over multiple bursty wireless channels," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 1106–1109.



Gene Cheung (M'00–SM'07) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1998 and 2000, respectively.

In August 2000, he joined Hewlett-Packard Laboratories Japan, Tokyo, where he is currently a Senior Researcher with the Multimedia Systems Architecture Group. His research interests include multimedia processing and networking, wireless networks, and combinatorial optimization.

Wai-tian Tan (M'01) received the B.S. degree from Brown University, Providence, RI, in 1992, the M.S.E.E. degree from Stanford University, Stanford, CA, in 1993, and the Ph.D. degree from the University of California, Berkeley, in 2000.

He joined Hewlett-Packard Laboratories, Palo Alto, CA, in December 2000 and is a member of the Streaming Media Systems Group. He worked for Oracle Corporation from 1993 to 1995. His research focuses on adaptive media streaming, both at the end-point and inside the delivery infrastructure.

Connie Chan received the B.S. degree in computer science from the University of British Columbia, Vancouver, BC, Canada, in 2004.

She was a student intern with Hewlett-Packard Laboratories Japan, Tokyo, Japan, during 2002. She was with Nokia, London, U.K., where she was involved with mobile product development from 2004 to 2006.